CS 325 Introduction to Game Design (Spring 2015)
Digital Assignment 2 — Digital Prototype

Due: Thursday, February 12th
(must be submitted through Blackboard before 12:00pm noon)
We will play the games in class!

Goals:

1. Increase your JavaScript programming skills.
2. Increase your interactive and animation programming skills.
3. Prototype an *original* idea or mechanic for a digital game.
4. Make a pitch/treatment/concept document.

Submission:

On Blackboard, submit the URL to your playable game hosted on GitHub. It will look like: http://yourname.github.io/game-name . Make sure that the repository is public.

Details:

Use Phaser and JavaScript to create a game prototype with the following components:

- Experimental or original gameplay
- Mouse or keyboard inputs that have meaningful impacts
- Movement and animation
- Sound effects

1. Create a new repository on GitHub. It isn't easy (via the GUI or via github.com) to fork your previous homework or the digital homework template (https://github.com/GMU-CS325/digital-template/), so you will probably find it easiest to simply create a new repository, clone it to your computer, and copy the digital-template files inside: https://github.com/GMU-CS325/digital-template/archive/master.zip.
2. Again, make your game on the `gh-pages` branch. Your assets (images, sound, etc.) go into the `assets` folder, your code goes into the `main.js` file, and your pitch document and how-to-play instructions go into `index.html`. You may split your code across multiple JavaScript files.
3. Push your changes to GitHub and verify that it runs from http://yourname.github.io/game-name. Your website will not appear there until your first push.
    3.1. While developing, you will want to test your game locally, without the added step of pushing to GitHub. For security reasons, web browsers won't let you simply double-click `index.html` to run it. You must run it from a local web server. If you have Python, simply enter the directory and type `python -m SimpleHTTPServer` (Python 2.x) or `python -m http.server` (Python 3.x). If you have `node.js`, you can easily install (`npm install http-server -g`) and run a local HTTP server (`http-server`). Then browse using Chrome to

your local web server: most likely http://localhost:8000/ or http://localhost:8080/.

3.2. It is possible to edit text files directly on github.com (the little pencil icon). This can be super-convenient, but make sure to sync the changes to your computer before editing there.

The code should be well-structured (using any encapsulation technique; see my JavaScript reference guide). Your pitch document (the `index.html`) must include a concise explanation of the inputs and their expected effects (how to play). Your code can borrow from any of the Phaser examples http://examples.phaser.io/ , but must differ in all of the following:

- Use different images
- Use different sounds
- Have different gameplay interactions and results

The resulting game should be fairly polished and the interactions should definitely appear intentional. You can use images and sprites from the internet, but cite where you got them in your pitch document. You must also cite which Phaser example most closely resembles your game.

This is your first digital prototype. This is also your first experience learning Phaser, so don't bite off more than you can chew. You'll be making more.

Pitch/Treatment/Concept Document Your `index.html` serves as your game's Pitch Doc (also known as a Treatment or Concept Document). There is currently placeholder text. Altogether, your `index.html` with the embedded prototype forms an interactive Pitch Doc, which is a pretty awesome way to "sell" your game to others, and great for your portfolio. Here is what it must include:

- High Concept: A one-sentence summary of your game.
- Player Experience Goals: What experience do you want players to have when playing your game?
- Genre: A single sentence that places the game within a game genre or a hybrid of genres.
- Game play: A paragraph describing the actions the player can perform, the system dynamics, and the core mechanic. Include a concise explanation of the prototype's inputs and their expected effects (how to play). You can also describe game play that is not in the prototype. You may include mock-up images for parts of the game not in the prototype.
- Prototype goal: What game mechanic is this prototype evaluating?
- Features: What are the unique aspects of your game that set it apart?
- Setting/Premise: A paragraph about the world your game is set in and who the characters are. What makes the game world and its occupants unique and

interesting? Do the tokens represent something? If the game has a backstory, put it here. If the game is abstract, then say so.

- Story: If the game has a story, summarize it in a paragraph. How will the dramatic tension interact with the gameplay tension?
- Target Audience: A single sentence that describes the demographic you're trying to reach.
- Play time: How long does your game take to play?
- Strategies: What player strategies do you expect will be effective at playing this game?
- Prototype Assets: Did you make your prototype assets from scratch? Did you borrow them? Cite your sources here.
- Prototype Closest Other Game: Which other game most closely resembles your game? If you are borrowing from a Phaser Example, you must say so here.

For reference, here are some example Pitch Docs:
http://jacobminkoff.files.wordpress.com/2010/02/wolvesextended-web.pdf
and here:
http://jacobminkoff.files.wordpress.com/2010/02/damnationextended-web.pdf

Grading:

Games will be given a number from 1 to 3 as the grade.

1. You will receive a 1 for satisfactory completion of the assignment.
2. You will receive a 2 for making something beyond “satisfactory”: an interesting gameplay experiment or high technical achievement.
3. Exceptional games (in gameplay, polish, or technical merit) will be awarded a 3.

NOTE: A 2 is a good score, equivalent to an A. There will be very few 3’s!